On environments as systemic exoskeletons: Crosscutting optimizers and antifragility enablers

Vincenzo De Florio

Abstract Classic approaches to General Systems Theory often adopt an individual perspective and a limited number of systemic classes. As a result, those classes include a wide number and variety of systems that result equivalent to each other. This paper presents a different approach: First, systems belonging to a same class are further differentiated according to five major general characteristics. This introduces a "horizontal dimension" to system classification. A second component of our approach considers systems as nested compositional hierarchies of other sub-systems. The resulting "vertical dimension" further specializes the systemic classes and makes it easier to assess similarities and differences regarding properties such as resilience, performance, and quality-of-experience. Our approach is exemplified by considering a telemonitoring system designed in the framework of Flemish project "Little Sister". We show how our approach makes it possible to design intelligent environments able to closely follow a system's horizontal and vertical organization and to artificially augment its features by serving as crosscutting optimizers and as enablers of antifragile behaviors.

1 Introduction

Classic approaches to General Systems Theory (GST) such as the ones introduced in [47] and [4] only consider a single, "horizontal" dimension. Boulding, for instance, classifies systems through "flat" systemic classes: a system may be regarded as a "Thermostat", or a "Cell", or

MOSAIC research group
University of Antwerp & iMinds research institute
Middelheimlaan 1, 2020 Antwerp, Belgium
E-mail: vincenzo.deflorio@gmail.com
E-mail: vincenzo.deflorio@uantwerpen.be

a "Plant", and so forth, though all systems belonging to any given class are no further differentiated.

A second aspect shared by classical general systems classifications is the individual and atomic perspective. In all behavioral classes introduced in [47] and all but one of those defined, for example, in [4] systems are considered as atomic, non-dividable elements. The only exception to this rule is Boulding's class of social organizations, which is defined as "a set of roles tied together with channels of communication", though it is no further analyzed.

A first contribution of this paper is the introduction of a novel approach to general systems classification. Following our approach, systems belonging to a same class are differentiated according to five major general characteristics. This introduces a "horizontal dimension" to system classification. A second component of our approach is introduced through the assumption that all systems should be considered as organizations of collective systems. Such a recursive definition translates into a *nested compositional hierarchy of sub-systems*, namely "a pattern of relationship among entities based on the *principle of increasing inclusiveness*, so that entities at one level are composed of parts at lower levels and are themselves nested within more extensive entities" [54]. From said assumption we derive the second, "vertical" classification dimension of our approach: at the same time, systems are considered as either systems-of-systems or network-of-networks, namely networks of nodes each of which may be another network. Each of those nodes is a system, eligible thus to be classified along our horizontal and vertical dimensions. Our stance is that a fair comparison of any two systems, say a and b, with respect to their features and emerging properties, should be done by considering those two dimensions, up to some agreed upon level of detail or scale.

A discussion of our classification is given in Sect. 2 while in Sect. 3 we briefly consider how our classification may be used in comparing the resilience of two systems.

The horizontal and vertical dimensions of our classification system are also one of the key characteristics of a distributed hierarchical organization called Fractal Social Organization (FSO). As in our general systems model, also FSO's [20, 23, 19] are a nested compositional hierarchy of nodes. Such nodes are building blocks of a complex organization and are called Service-oriented Communities [21, 22].

Section 4 briefly recalls the major elements of FSO's. A second contribution of this paper is the idea to make use of the FSO organization to design servicing infrastructures mimicking a system's vertical dimension and interfacing its "horizontal" components [27, 16]. This may be used, e.g., to create intelligent environments able to empower a community subjected to a natural or human-induced disaster [26].

Our conclusions and a view to future work are finally stated in Sect. 5.

2 Two dimensions of system classification

In Sect. 1 we observed how traditional GST's mostly define "flat" classes of systems, and that said systems are often considered as individual, atomic (i.e., non-decomposable) systems. A reason for this is possibly that traditional theories are based on one or more *systemic touchstones*, which we defined in [14] as

> "privileged aspects that provide the classifier with 'scales' to diversify systems along one or more dimensions."

The classic term to refer to such aspects is *gestalt*, namely the "essence or shape of an entity's complete form" [31]. The accent is thus on a system's salient traits rather than on its architectural composition or its organizational design.

An important consequence of gestalt-based classification methods is the fact that they function as *models*: they highlight certain aspects or features of a system while hiding others. As an example, the behavioral gestalt introduced in [47] only focuses on

> "the examination of the output of the [system] and of the relations of this output to the input. By output is meant any change produced in the surroundings by the [system]. By input, conversely, is meant any event external to the [system] that modifies this [system] in any manner."

This results in generic classes that include very different systems—for instance natural systems, artificial, computer-based systems, and business bodies[1]. From a practical point of view, systems in a class are considered as equivalent representatives of their class—as it is the case in equivalence classes in algebra [49]. This is exemplified by the relation of "Boulding-equivalence" introduced in [10].

Definition 1 (Systemic classes) Given any GST T defining $n > 1$ classes of systems according to a given gestalt g, we shall call T-equivalence the equivalence relation corresponding to the n classes of systems. Those classes shall be called "systemic classes according to T and g", or, when this may be done without introducing ambiguity, simply as "systemic classes."

Moreover, traditional system classifications pay little or no attention to the collective nature of systems. In other words, systems are mostly considered as individual, monolithic entities instead of the result of an organization of parts, each of which is in itself another system.

As we have shown in [15], this translates in a *partial order* among systems: systems may be practically compared with one another—for instance, as of their intrinsic resilience [17]—only if they belong to different systemic classes. There is no easy way to tell which of two Thermostats, or for instance two Cells[2], is better suited to manifest a given emerging property.

In what follows we propose to tackle this problem by considering two "dimensions":

- A "horizontal" dimension, regarding the system as an entity resulting from the organization of a number of peer-level individual components.
- A "vertical" dimension, regarding the system as a collective entity resulting from the social organization (*sensu* [4]) and cooperation of a number of organs, each of which is also socially organized into a collection of other organs [18].

2.1 Horizontal dimension of system classification

Our starting point here is the conjecture that most of the classes introduced in GST's may be described in terms of the five components of the so-called MAPE-K loop of autonomic computing [32], corresponding respectively to

---

[1] Boulding's class, for instance, include among others "Clockworks", "Thermostats", "Cells", "Plants", "Animals", and "Transcendental Systems", which are generic names that may refer to systems of any nature.

[2] As already mentioned, Thermostat and Cell are the names of two classes of the Boulding-equivalence relation.

$\mathcal{M}$: the ability to perceive change;

$\mathcal{A}$: the ability to ascertain the consequences of change;

$\mathcal{P}$: the ability to plan a line of defense against threats deriving from change;

$\mathcal{E}$: the ability to enact the defense plan being conceived in step 3;

$\mathcal{K}$: the ability to treasure up past experience and continuously improve, to some extent, abilities $\mathcal{M}$–$\mathcal{E}$.

Definition 2 (Systemic features) As we have done in [15], in what follows we shall refer to abilities $\mathcal{M}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{E}$, and $\mathcal{K}$ as to a system's *systemic features.*

As an example of the expressiveness of a system classification based on the above systemic features, it is easy to realize that the systemic class of purposeful, non-teleological systems [47], corresponding to Boulding's Thermostats, can also be interpreted as the class of those systems that are characterized by very limited perception ($\mathcal{M}$), analytical ($\mathcal{A}$), and operational ($\mathcal{E}$) quality and by the absence of planning ($\mathcal{P}$) and learning ($\mathcal{K}$) ability. Another example is given by the systemic class of extrapolatory systems, which roughly corresponds to Boulding's class of Human-Beings. Systems in this class possess complex and rich systemic features $\mathcal{M}$–$\mathcal{K}$.

As already mentioned, an intrinsic problem with systemic classes is that all of the systems in a class are evened out and equalized. Obviously this is problematic, because systemically equivalent systems may be in fact very different from each other. Two Thermostats may base their actions on different context figures—think for instance of an accelerometer and a gyroscope when used for fall identification [24,30]. Two Human-Being systems may have different analytical, planning, or learning features due to, e.g., different design trade-offs[3].

Mapping the existing GST's onto the five systemic features allows for a finer differentiation if we further decompose each class into sub-classes. A way to do this has been described, for perception and analytical organs, in [10] and for planning organs in [12].

The idea is to either detail the quality of a systemic feature or to identify the systemic class of the corresponding organs.

For perception, the quality of $\mathcal{M}$ is made explicit—to some extent—by specifying which subset of context figures is perceived by $\mathcal{M}$. Notation "$\mathcal{M}(M)$" is then used to state that perception is restricted to the context figures specified in set M. In next subsection we show how this makes it possible to use simple Venn di-

agrams to compare the perception feature in systems and environments.

2.2 Perception

Let us consider any two systems a and b, respectively characterized by $\mathcal{M}(A)$ and $\mathcal{M}(B)$. There can be two cases: either

$$(A \subset B) \vee (B \subset A) \tag{1}$$

or otherwise, namely

$$(A \not\subset B) \wedge (B \not\subset A). \tag{2}$$

As we showed in [12], if (1) is true and in particular $A \subseteq B$, then we shall say that b is endowed with a greater perception than a. Notation $a \prec_{\mathcal{P}} b$ will be used to express this property. Likewise if (1) and $B \subseteq A$ then we shall have that $b \prec_{\mathcal{P}} a$. This is exemplified in Fig. 1(b), in which

$$A \subseteq B \subseteq M, \tag{3}$$

the latter being the set of all the possible context figures. Clearly no system m such that $\mathcal{M}(M)$ exists, though we shall use of it in what follows as a reference point—a hypothetical system endowed with "perfect" perception and corresponding to the "all-seeing eye" of the *monad*, which "could see reflected in it all the rest of creation" [36].

Expression (3) tells us that a, b, and m are endowed with larger and larger sets of perception capabilities. Expression $a \prec_{\mathcal{P}} b \prec_{\mathcal{P}} m$ states such property.

A similar approach may be used to evaluate the environmental fit of a given system with respect to a given deployment environment. As an example, Fig. 1(a) may be also interpreted as a measure of the perception of system a with respect to the measure of perception *called for* by deployment environment b. The fact that $B \setminus A$ is non-empty tells us that a will not be sufficiently aware of the context changes occurring in b. Likewise $A \setminus B \neq \varnothing$ tells us that a is designed so as to be aware of figures that will *not* be subjected to change while a is in b. The corresponding extra design complexity is (in this case) a waste of resources in that it does not contribute to any improvement in resilience or survivability.

Finally, Venn perception diagrams may be used to compare environments with one another. This may be useful especially in ambient intelligence scenarios in which some control may be exercised on the properties of the deployment environment(s).

Estimating shortcoming or excess in a system's perception capabilities provides useful information to the "upper functions" responsible for driving the evolution

---

[3] Explanations and examples of those trade-offs in natural systems can be found, e.g., in [42,41,61,43].
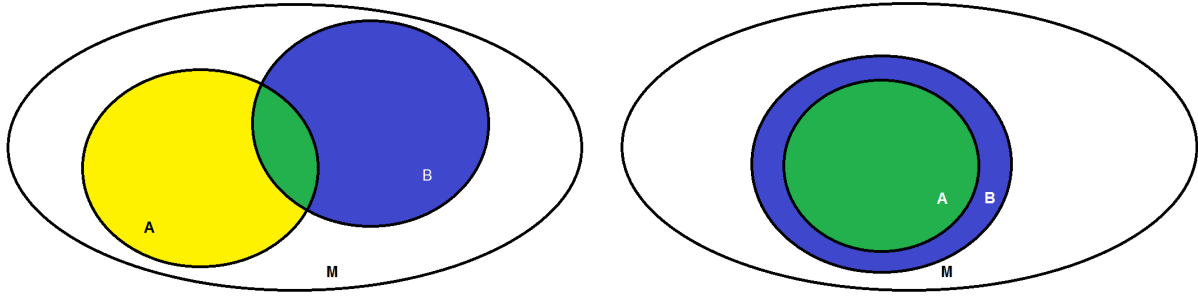
(a) Perception of systems a and b with respect to that of hypothetical perfect system m in the case of (2). The intersection region represents the context variables perceived by both a and b.

(b) Perception of a and b when (1) is valid. In this case we can state that $a \prec_{\mathcal{P}} b \prec_{\mathcal{P}} m$: the perception feature of a is less than b's, which in turn is less than m's.

Fig. 1 Venn diagrams are used to reason about the systemic features of perception in two systems.

of that system. Such functions may then make use of said information to perform design trade-offs among the resilience layers. As an example, a system able to do so may reduce its perception spectrum and use the resulting complexity budget to widen both its $\mathcal{A}$ and $\mathcal{P}$ systemic features.

2.2.1 Limitations of our approach

Although effective as a secondary classification system, our approach is also a model—in other words, a simplification. In particular, reasoning merely in terms of subsets of context figures underlies the unlikely assumption of an at-all-times perfect and at-all-times reliable perception organ. Furthermore, our approach does not take into account the influence that other organs may have on the perception organ[4].

2.3 Other systemic features

The above approach based on Venn diagrams cannot be applied to systemic features $\mathcal{A}$–$\mathcal{K}$. An alternative approach was suggested in [12]. The idea is to select a GST T and "label" each organ with its systemic class in T. This allows a finer classification to be obtained and a greater differentiation of systems in the same systemic class.

An exemplary way to apply this method is shown in [15] by making use of the classic behavioral method. Thus for instance the organ responsible for planning responses to context changes—corresponding thus to systemic feature $\mathcal{P}$—may be characterized as belonging to,

e.g., the "Thermostat" systemic class. As an example, our adaptively-redundant data structures [11] belong to the systemic class of predictive mechanisms, although their $\mathcal{P}$ organ belongs to the simpler class of purposeful, non-teleological systems.

In certain cases, instead of a GST, one could use an existing classification peculiar of a given systemic feature. Lycan, for instance, suggests the existence of at least eight apperception classes [37] (namely, eight $\mathcal{A}$ classes).

2.4 Vertical dimension of system classification

> "The Internet is a system —
> and any system is an internet."
>
> https://goo.gl/WTnvLD

As we already mentioned, a classic assumption shared by several GST's is that of describing systems from an individual perspective. Our "horizontal" classification proposes a first solution to this deficiency by providing a top level view to a system's organization. By exposing the main organs $\mathcal{M}$–$\mathcal{K}$ we provide a more detailed information about the nature and features of the system at hand.

Our vertical classification goes one step further. It does so by regarding systems as collective entities resulting from the social organization (*sensu* [4]) and cooperation of a number of organs, each of which is also socially organized into a collection of other organs. As in Sect. 2.1 systems were exposed as systems-of-systems, similarly here we model systems as network-of-networks. Better, systems are interpreted here as networks of nodes, each of which is in itself another network of nodes. As discussed in Actor-network Theory [34], each node

---

[4] As illustrated in, e.g., [5], perception may be misled by higher functions; as an example, the analytical organ may provide an interpretation of the ongoing facts that may lead the perception organ into "concealing" certain facts or over-exposing others.

"blackboxes" and "individualizes" its network by assuming the double identity of individual and collective system—a concept that finds its sources in the philosophy of Leibniz [14].

Being a system, each node is eligible to belong to a systemic class. The horizontal classification introduced in Sect. 2.1 may therefore be applied: a given node may for instance behave as an object [47] and thus be perception-free; or it may be a Thermostat or a "Servomechanism" (thus with limited perception and no analytic functions); or it may be an organ, as it is the case in Boulding's Cells. In such a case, it may be endowed with perception and limited analytical capabilities. Moreover, it may be an organism (a Plant or an Animal) and be endowed with extended perception, some analytical capabilities, and limited planning capabilities. At the top of the scale, it may be a self-conscious system (Boulding's Human-Beings) and rank high on all the systemic features.

Definition 3 (Systemic level) Given any system vertically classified into a network of nodes, we shall refer to each set of nodes that are peer levels as to a *systemic level*.

An example of systemic level is given by the top level view resulting from our horizontal classification. Another example is the very root of the vertical classification, namely the individual, "holistic" representation of the system—although of course in this case the systemic level is a singleton.

2.5 Preliminary conclusions

In this section we have introduced a horizontal and a vertical system classification as a tool to further differentiate systems belonging to a same GST class. By making use of our proposed classifications any system is organized both vertically and horizontally: vertically, as a network of nodes; and horizontally, as an organization of peer-level organs corresponding to the system's five systemic features.

We deem important to highlight how, by means of our classifications, systems expose their structure of complex networks of systems-within-systems, or equivalently of network-of-networks. This translates into a Matryoshka-like structure corresponding to the class of networks known as **nested compositional hierarchies** (NCH).

NCH have been defined in [54] as "a pattern of relationship among entities based on the principle of increasing inclusiveness, so that entities at one level are composed of parts at lower levels and are themselves nested within more extensive entities". The class of NCH organizations is widespread in natural systems because of its straightforward support of modularity—in turn, an effective way to deal with complexity and steer evolvability [58]. Further discussion on this may be found, e.g., in [16].

Finally, we remark how vertical organization and NCH produce a *fractal organization* of parts in a variety of levels, or scales. In natural systems those scales range from the microscopic, sub-atomic to the macroscopic level as typical of, e.g., biological ecosystems. When classifying systems in order to compare their systemic characteristics a trade-off shall be necessary in order to limit the vertical expansion to a practically manageable number of levels.

More information and an example of fractal organization are provided in Sect. 4.1 and Sect. 4.2.

3 Making use of our classification system to assess and compare resilience

Let us consider two examples:

1. A bullet passing through the body of a living being. Such a traumatic event directly affects a number of organs and systems of that being. Interdependence among organs and systems is likely to lead to cascading effects that may in turn lead to severe injuries or the loss of life.
2. As a second example, let us consider the case of a hurricane hitting a region. Catastrophic events such as this one typically ripple across the network-of-nodes triggering the concurrent reactions of multiple crisis management organizations [9,46].

The above two cases exemplify what we conjecture may be a "general systems law": any catastrophic event that manifests itself within a system's boundaries creates a critical condition that *crosscuts* all of that system's levels and nodes, with consequences that can affect the nodes that are directly hit as well as those depending on them. Consequences may ripple through the boundaries of the system and lead to chains of interconnected local failures possibly bringing to general system failure.

In fact, catastrophic events such as the two exemplified ones reveal a system's true nature and organization—as litmus paper does by unravelling the pH value of a chemical solution [2]. The illusion of an "in-dividual" (non-divisible) system is shuttered and replaced by the awareness of the fragmented nature of the system as a system-of-systems and a network-of-networks.

The adoption of a horizontal and a vertical system classification offers in this case a clear advantage, in

that it provides a view to the actions that may be expected from each of the involved constituent systems. Depending on each system's systemic level, the reaction to the catastrophic event might include different flavors of perception steps; analytical steps; planning steps; reaction plan execution steps; and knowledge management steps (namely, knowledge feedback and its persistence). Conditional "might" is used here because, as mentioned already, not all the involved systems may have a complete set of systemic features and the corresponding organs may have different systemic classes. Thus for instance the catastrophic event and its ripples will only be perceived by systems whose perception organs include the context figures related to that event. As another example, a $\mathcal{P}$ organ may produce a response plan ranging from predefined responses up to reactive, extrapolatory, and even antifragile behaviors [17].

Other factors may play a key role in local and overall responses to catastrophic events. Those responses may depend, e.g., on the quality and performance of the involved organs. Said quality may be modeled as a dynamic system and expressed in terms of fidelity and its fluctuations (called "driftings" in [25,13]). Moreover, as responses call for energy and energy being often a limited and valuable commodity, responses enacted by some nodes are likely to exact energy from other nodes[5].

Other important factors in the emergence of resilience are given by what we commonly refer to as "experience" and "wisdom", which correspond to systemic feature $\mathcal{K}$. Those factors are in some cases of key importance, as they may lead to situations in which two identical systems reach very different degrees of resilience [60].

A final and very important aspect that is not considered by our classification system is given by harmony and cohesion between the "parts" and the "whole". This is exemplified by the famous apologous that Menenius Agrippa gave the commons of Rome during the so-called "Conflict of the Orders" [40]. In his speech Agrippa imagines a disharmony among the parts of the human body, with "busy bee" organs complaining about the less active role played by other organs. Because of the discord, the more active parts undertake a strike, whose net result is a general failure because in fact all parts are necessary and concur to the common welfare according to their role and possibility[6]. In other words, disharmony is a disruptive force that

breaks down the whole into its constituent parts. Resilience may very well be affected in the process, as exemplified by a nation unable to effectively respond to an attack because of the lack of identification of its citizens with the state.

3.1 Resilience as an interplay of opponents

In our previous work [17] we discussed resilience as the emerging result of a dynamic process that represents the dynamic interplay between the behaviors exercised by a system and those of the environment it is set to operate in. With the terminology introduced in this paper we may say that resilience is the result of the effects of an external event on a system's horizontal and vertical organization. The external event manifests itself at all systems and networks levels and activates a response that is both individual and social. As we conjectured in the cited reference, game theory (GT) [28] may provide a convenient conceptual framework to reason about the dynamics of said response. GT players in this case are represented by nodes, while GT strategies represent the plans devised by the nodes' $\mathcal{P}$ organs. As suggested in our previous work, a way to represent the strategic choices available to the GT players is to classify them as behaviors. As an example, if node n is able to exercise extrapolatory behaviors, then n may in theory choose between the following strategies of increasing complexity: random; purposeful/non-teleological; teleological/non-extrapolatory; or extrapolatory [47]. In practice, the choice of the strategy shall also be influenced by some "energy budget" representing the total amount of consumable resources available system-wide to enact the behaviors of all nodes. Said energy budget would then serve as a global constraint shared by all of the nodes of the system across both the horizontal and vertical organizations.

GT payoffs could then be associated to the possible exercised behaviors, with costs (in terms of consumed energy budget resources) proportional to the complexity of the chosen behavior.

It seems reasonable to foresee that the adoption of GT as a framework for discussing the resilience of systems classified according to our approach shall require the definition of nested compositional hierarchies of payoff matrices—sort of interconnected and mutually influencing payoff "spreadsheets".

---

[5] For example, "inner" systems' action may deprive outer systems of their resources; and likewise outer systems' decisions may lead to poor choices affecting the resources and the operational conditions of inner nodes.

[6] See for instance [36]: "There is always in things a principle of determination which must be sought in maximum and minimum; namely, that the greatest effect should be produced with the least expenditure, so to speak."

4 An intelligent environment based on our system classification

In Sect. 3 we have considered resilience, interpreted as the outcome of a *conflict* between two opponents. We have shown that our system classification allows said conflict to be detailed within the systems boundaries along their vertical and horizontal dimensions.

A dual consideration can be made by considering other emerging properties—for instance, performance, safety, and quality of experience. An intelligent environment may be designed with the aim to assist a system to achieve its design goals by structuring it *after* the horizontal and vertical classification of that system. One such system is the middleware designed in the framework of project "Little Sister". In what follows we briefly introduce some elements of that project that are useful to the present discussion and then we suggest how the architecture of the LS middleware may facilitate the expression of optimal services combining emerging properties such as the above mentioned ones.

4.1 Little Sister

Little Sister (LS) is the name of a Flemish ICON project financed by the iMinds research institute and the Flemish Government Agency for Innovation by Science and Technology. The project run in 2013 and 2014 and aimed to deliver a low-cost telemonitoring [38] solution for home care. Cost-effectiveness was sought by replacing expensive and energy-greedy smart cameras with low-resolution cameras based on battery-powered mouse sensors [3].

The LS software architecture is exemplified in Fig. 2. As suggested by the shape of the picture, LS adopts a fractal organization in which a same building block—a web services middleware component—is repeated across the scales of the system. In fact the vertical classification of the LS service is, in a sense, *revealed* through the fractal organization [33, 59, 55, 56] of the LS software architecture:

- Atomic constituents are grouped into a "level 0" of the system. Those constituents are wrapped and exposed as manageable web services that represent a periphery of $\mathcal{M}$ and $\mathcal{E}$ nodes.
- Said web services constitute a first level of organs that manage the individual rooms of a smart house under the control of a middleware component responsible for systemic features $\mathcal{A}$ and $\mathcal{P}$.
- Individual rooms are also wrapped and exposed in a second level under the control of the same middleware component, here managing a whole smart house.

- The scheme is repeated a last time in order to expose smart house services, also under the control of our middleware component. This third level is called smart building level.

No systemic feature $\mathcal{K}$ is foreseen in LS.

As evident from the above description, the LS system represents a practical example of our horizontal and vertical classification. The levels of the LS software architecture allow services to be decomposed into

- Low-level services for context change identification;
- Medium-level services for situation identification [62];
- High-level services for overall system management,

which naturally leads to the choice of a three-level vertical classification.

4.2 A fractally organized intelligent environment

As mentioned at the beginning of this section, awareness of a system's horizontal and vertical classification may be exploited to create an environment reflecting the structure of that system and designed in order to provide assistive services to that system. In what follows we provide an example of such an environment, implemented through the LS middleware.

As the system is structured into three levels so also our middleware is organized on three levels. A same middleware module resides in the three environments that represent and host the nodes of each level: rooms, houses, and building, and corresponding respectively to levels 1–3 in Fig. 2. The middleware wraps sensors and exposes them as manageable web services. These services are then structured within a hierarchical federation [45]. More specifically, the system maintains dedicated, manageable service groups for each room in the building, each of which contains references to the web service endpoint of the underlying sensors (as depicted in level 0 and 1 in Fig. 2). These "room groups" are then aggregated into service groups representative of individual housing units. Finally, at the highest level of the federation, all units pertaining to a specific building are again exposed as a single resource (level 3). All services and devices situated at levels 0–3 are placed within the deployment building and its housing units; all services are exposed as manageable web services and allow for remote reconfiguration.

By exposing the sensors as manageable web services and by means of a standardized, asynchronous publish-and-subscribe mechanism [44] the middleware "hooks" onto the system's perception and executive organs—namely those organs corresponding to systemic features $\mathcal{M}$ and $\mathcal{E}$. All status and control communication is thus
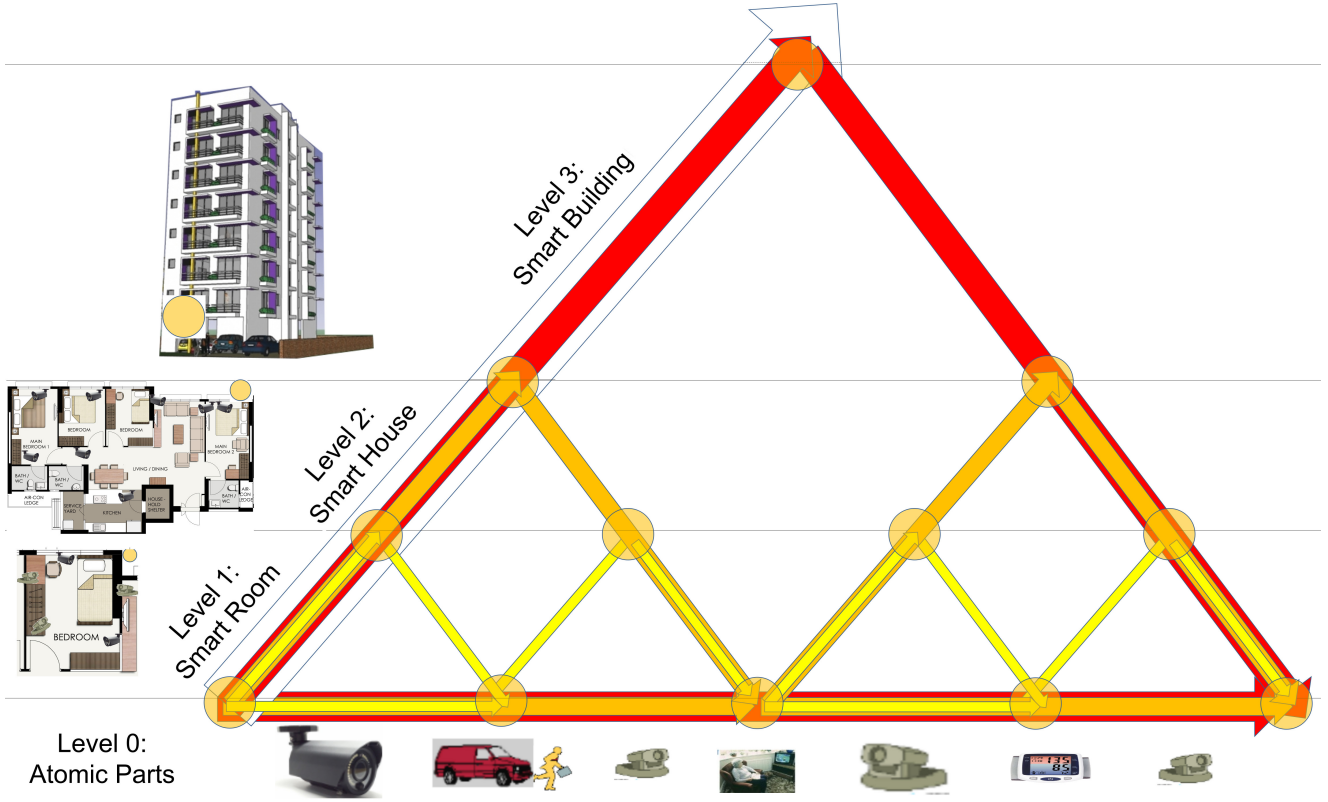
Fig. 2 Exemplification of the LS software architecture.

transparently received by the middleware, which checks whether the received information calls for functional adjustments or if it represents a safety-critical situation requiring a proper response.

Each response is managed by the middleware through a protocol that requires the cooperation of "agents" (system nodes). As in data-flow systems [48,57] it is the presence of the input data that "fires" an operation, likewise in LS protocols it is the presence of all the required roles that enables the launch of a protocol. For this reason we refer to our approach as to a *role-flow scheme.*

Said role-flow scheme of the LS middleware is a simplified version of the more general strategy introduced in [20,21], in which nodes publish a semantic description of the roles they may play and the services they may offer. Semantic matching is then used in the enrollment phase [51].

The above sketched distributed organization, in which a same building block is repeated in a nested compositional hierarchy of nodes, is known in the literature as a *fractal organization* [33,59,55,56]. "Canon" is the term used to refer to a fractal organization's building block. Each node of the hierarchy hosts a canon—which in the case at hand is implemented by our middleware module.

It is important to highlight how the canon at level i is both a node of that level and a node of level $i+1$ (if i is not the top level). As a node of level i, the canon plays the role of that level's "controller" by executing the role-flow scheme. At the same time, canon i represents and "punctualizes" [35] the whole level i into a single level $i + 1$ node.

A peculiarity of the fractal organization of our middleware is the interoperability and cooperation between its levels—a feature that is achieved through the concept of *role exception*. When middleware module at level i does not find all the roles required to launch a protocol, it declares an exception: being also a node of level $i + 1$, its status and notifications are transparently published and received by the middleware module at level $i + 1$. The latter thus becomes aware of a level i protocol that is missing roles. Missing roles are thus also sought into the parent node, and from there into the parent's parent node, and so on.

A consequence of this strategy is that roles are first sought in the level where a "need" has arisen; only when that level fails to answer the need, the hierarchy is searched in order to complete the enrollment and launch the protocol. The result of this strategy is a new, trans-hierarchical "temporary organ", consisting of the

nodes in any level of the hierarchy that best-match the need at the time of enrollment.

Since the new organ includes nodes from multiple levels of the network-of-networks, we call the new organ a social overlay network (SON). Fractal social organization (FSO) is the name we gave to a fractal organization implementing the above strategies [20].

4.2.1 Adaptive dimensioning of response protocols

The same algorithm employed for the adaptively-redundant data structures mentioned in Sect. 2.3 was adopted for the LS middleware. In what follows we briefly describe that algorithm.

As mentioned above, the middleware becomes timely aware of the state of the LS system. This includes the definition of the current "situation". Situations [62] range from low-level context changes pertaining to the state of devices (for instance, a sensor's battery level reaching a given lower threshold) up to high-level, human-oriented conditions and events. An example of the latter case is situation s_1 = "the resident has left her bed during the night and is moving towards the kitchen". Another example is s_2 = "the resident is sleeping in her bed". In general, different situations call for different reaction protocols, in turn calling for a different amount of nodes and resources. For instance, it may be sensible to appoint more resources to situation s_1 than to s_2. Our algorithm implements an adaptive dimensioning strategy that estimates the amount of nodes best-matching the current situation with minimal impacting on the system's design goals.

In the absence of activity and when the current situations are assessed as relatively stable and safe—as it appears to be the case in situation s_2—the middleware gradually decreases its requirements down to some minimum threshold. This threshold level is estimated beforehand so as to still guarantee prompt reaction as soon as variations are detected in the ongoing scenarios.

In a sense, the LS middleware tracks the activity of the residents closely *imitating* their behaviors: when a resident, e.g., sleeps, the corresponding LS entity also goes-to-sleep (or better, it goes to low consumption mode). On the contrary, when the residents awake or are in need, the LS entity also goes to full operational mode.

As already mentioned, the gradual adjustments of the LS operational mode is based on an algorithm of autonomic redundant replicas selection. At regular time steps the middleware component responsible for the current level checks whether the current allocation was overabundant or underabundant with respect to the ongoing situation. In the former case—namely if "too many" resources were employed, the container selects some of the enrolled nodes and "frees" them. In the latter case, either a "better" selection of the same amount of fractals is attempted, or new fractals are enrolled, or both, by following the strategy depicted in [7,8]. A "distance-to-failure" function is computed at each step to measure how the current configuration matched the current situation. The value of this function determines overabundance vs. underabundance and the corresponding decrease vs. increase of the employed resources.
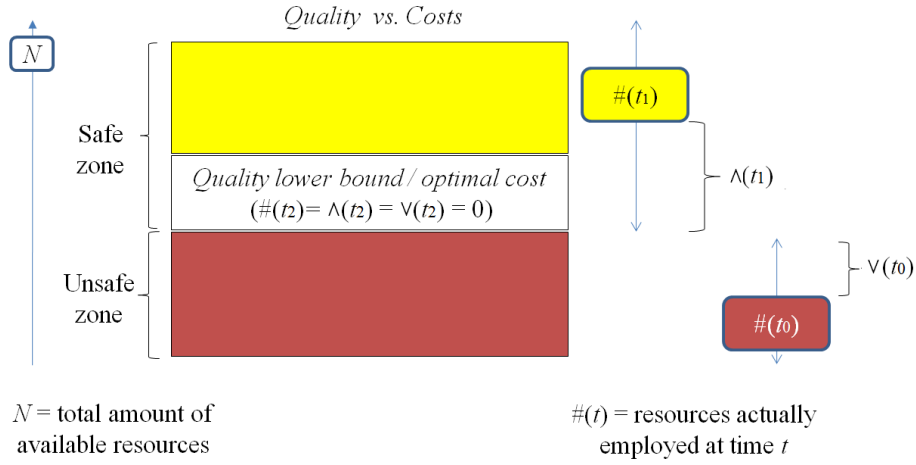
The logic of this algorithm is graphically represented in Fig. 3. In such picture, N is the total amount of nodes available (e.g., 10 sensors deployed in different positions in a resident's bedroom) and $\#(t)$ is the amount of "fired" (namely, activated) sensors. If we assume that the current situation, say s, will not change during a certain observation interval T (because, for instance, the resident is sleeping in her bed), then during T we will have two stable "zones" corresponding to the different allocation choices enacted by LS.

- The unsafe zone is depicted as a red rectangle and represents choices corresponding to resource undershooting: here too few nodes were allocated with respect to the situation at hand. For any $t \in T$, function $\vee(t_0)$ tells us how big our mistake was at time t_0: how far we were at t_0 from the minimal quality called for by s.
- The safe zone is given by the the white and the yellow rectangles.
  - In the yellow rectangle the allocation was overabundant: too many resources were allocated (resource overshooting). For any $t_1 \in T$ function $\wedge(t_1)$ tells us how large our overshooting was at time t_1. It also represents how far we were from the unsafe zone.
  - The white rectangle represents the best choice: no overshooting or undershooting is experienced, which means that the allocation matches perfectly situation s. Here $\vee(t_2) = \wedge(t_2) = 0$.

The above mentioned "distance-to-failure" is then defined, for any t, as

$$\mathrm{DTOF}(t) = \frac{\vee(t)}{N}. \tag{4}$$

The allocation strategy of LS is based on tracking the past values of DTOF in order to estimate the "best" allocation of resources for next step. Regrettably, no implementation of the above design was completed in the course of project LS, although a study of the performance of our strategy is ongoing [6], with preliminary results available in the above cited papers.

Quality vs. Costs

N = total amount of
available resources

$\#(t)$ = resources actually
employed at time t

Fig. 3 LS optimization is based on measuring resource overshooting (function $\wedge$) and undershooting (function $\vee$) and adjusting resource allocation accordingly. The picture shows the three possible system states: in t_0 the system experiences resource undershooting; in t_1, resource overshooting; and in t_2 it reaches an optimal resource expenditure.
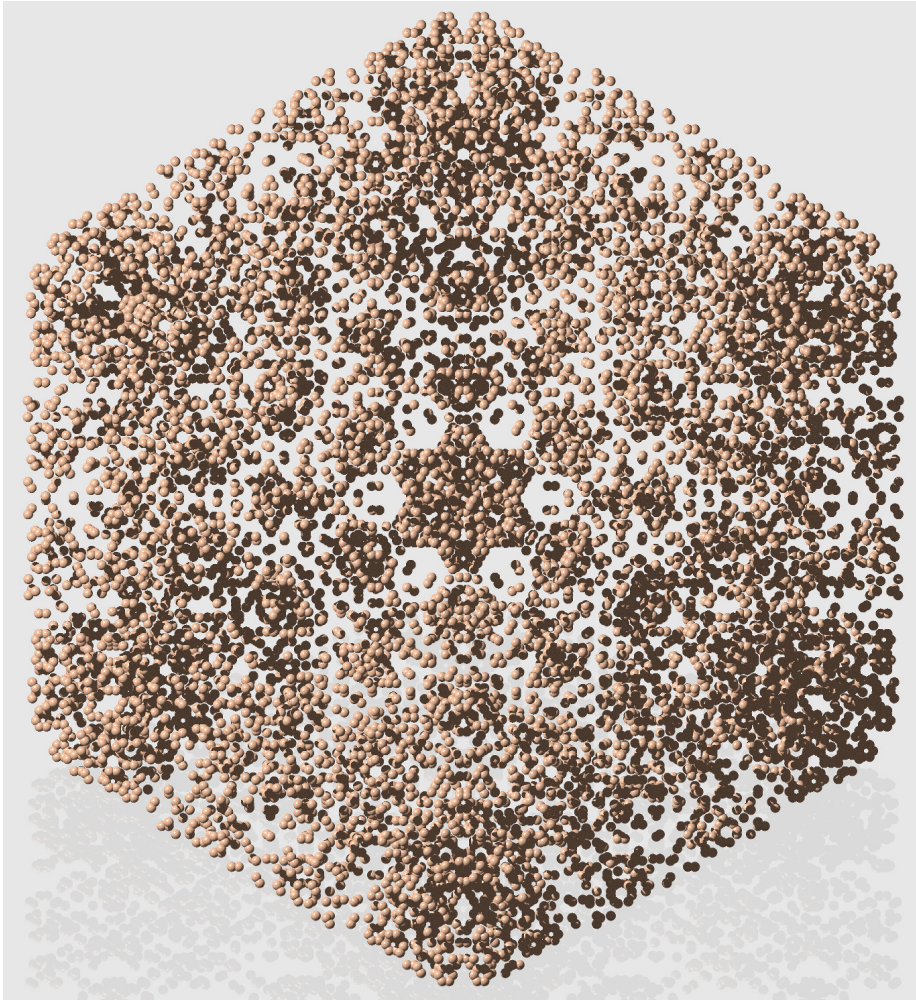
Fig. 4 The graph represents the set of all the social overlay networks that may emerge from an FSO with 9 nodes and 5 roles (roles 0–4). In this case the FSO consists of three nodes able to play role 0; three nodes able to play respectively roles 1–3; and three roles able to play role 4. The graph was produced with the POV-Ray ray tracing program [29].

Figure 4 shows a three-dimensional representation of the space of all possible SON's that can originate from an exemplary FSO.

4.3 Environments as crosscutting optimizers and antifragility enablers

We now briefly discuss the approach exemplified in Sect. 4.2 by considering environments as crosscutting optimizers as well as enablers of antifragile behaviors.

4.3.1 Environments as crosscutting optimizers

We conjecture that environments such as the one we have just sketched may function as *crosscutting optimizers*: assisting environments able to rapidly communicate awareness and wisdom from one level to the other of an assisted system. This is made possible by means of the mechanisms implemented by our FSO: exception, role-flow, and SON. In FSO, unresolved local events are transferred automatically to the higher levels of the organization. Local decisions and reactions are then exposed to the higher levels, and vice-versa: actions and decisions occurring in the higher levels of the system may thus be perceived and analyzed by the "inner systems", allowing those systems to understand the local consequences of "global" actions.

We conjecture that this may result in perception failures avoidance, reduced reaction latency [39,1], increased agility, and avoidance of single-points-of-congestion. Furthermore, the FSO enrollment does not discriminate between institutional and non-institutional nodes. This encourages participation and collaboration and avoids community resilience failures such as the ones experienced in the recovery phase after the Katrina and Andrew Hurricanes. The same non-discriminative nature makes it possible for unnatural distinctions between, e.g., primary, secondary, and tertiary users, to be avoided [52].

4.3.2 Environments as antifragility enablers

As a second conjecture, we believe that environments based on our FSO may also function as *antifragility enablers*[7].

---

[7] Antifragility is the term introduced by N. N. Taleb in [53] to refer to systems that are able to *systematically* "enhance the level of congruence or fit between themselves and their surroundings" [50]. Quoting from Professor Taleb's book,

> "Antifragility is beyond resilience or robustness. The resilient resists shocks and stays the same; the antifragile gets better."

An analysis of elastic, resilient, and antifragile behaviors was proposed in [17].

As we have already remarked, the LS middleware does not provide a complete implementation of FSO. In particular, it does not foresee any component responsible for the $\mathcal{K}$ systemic feature. A major consequence of this is that the FSO enrollment in LS is *memoryless*: protocols are started from scratch, taking no account of their past "history". Aspects such as the performance of a node as "role player" in the execution of a protocol; the trustworthiness manifested by that node; the recurring manifestation of a same SON; as well as its performance as executive engine for a protocol; were not considered in the LS design. Of course nothing prevents to design a FSO in which the above and additional aspects are duly considered. As an example, enrollment scores, telling which nodes best played a role in a given SON, may be implemented by making use of algorithms of gradual rewarding and penalization such as the ones described in [7,8,6]. Those very same algorithms, applied at a different *level*, may be used to gain wisdom as to the best-matching solutions. Proactive deployment of the best-scoring SON's across the scales of the FSO may enhance its effectiveness in dealing with, e.g., disaster recovery situations. Moreover, the resurfacing of the same transient SON's may lead to *permanentification*, namely system reconfigurations in which new permanent nodes and levels manifest themselves. In other words, by means of the above and other antifragile strategies the system and its vertical organization may *evolve* rather than *adapt* to the conditions expressed by a mutating environment.

5 Conclusions

In the present work we have proposed to augment existing GSTs by making use of a horizontal and a vertical dimension. This introduces systemic subclasses that make it possible to further differentiate systems belonging to the same GST class. We have shown how this allows for a finer comparison of systems with respect to their ability to achieve their intended design goals. In particular we have shown how to make use of our classification approach to assess the resilience exhibited by a system when deployed in a target environment. Building on top of our previous work on resilient behaviors, here we have further discussed resilience as a property emerging from an interplay of the behaviors exercised by two opponents.

As a dual argument, here we have also considered properties emerging from interplays of "opposite sign"—namely, interplays between a system and an *assisting* (rather than an opposing) environment. We have discussed how our classification approach allows for the

creation of an environment mimicking a system's horizontal and vertical structure. By doing so, the assisting environment realizes a "systemic exoskeleton" of sorts, which is able to interface with that system's organs and artificially augment its analytical, planning, and knowledge systemic features. In particular we have shown how FSO and its concepts of exception, role-flow, and SON, realize inter-organizational collaboration between nodes residing in any of the levels of the system organization.

Future work will include the simulation of scenarios in which the environment plays either the role of opponent or that of assistant, as discussed in this paper. Preliminary results have been already obtained by simulating ambient assistive environments [23, 24]. A theoretical discussion of resilience in the framework of Game Theory is also among our plans.

Acknowledgments.

References

1. Adair, B.: 10 years ago, her angry plea got hurricane aid moving. St. Petersburg Times. Available online at http://www.sptimes.com/2002/webspecials02/andrew/day3/story1.shtml (20 August 2002) 11
2. Advameg: How products are made: Litmus paper (2015). Retrieved on 2015-08-07 from http://www.madehow.com/Volume-6/Litmus-Paper.html 5
3. Anonymous: Introducing the Silicam IGO (2013). URL http://www.silicam.org/docs/flier.pdf 7
4. Boulding, K.: General systems theory—the skeleton of science. Management Science **2**(3) (1956) 1, 2, 4
5. Briscoe, R., Grush, R.: Action-based theories of perception. In: E.N. Zalta (ed.) Stanford Encyclopedia of Philosophy. The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University, Stanford, CA 94305-4115 (2015). URL http://plato.stanford.edu/entries/action-perception/ 4
6. Buys, J.: Voting-based approximation of dependability attributes and its application to redundancy schemata in distributed computing environments (2015). Unpublished draft 9, 11
7. Buys, J., De Florio, V., Blondia, C.: Towards context-aware adaptive fault tolerance in soa applications. In:

8. Buys, J., De Florio, V., Blondia, C.: Towards parsimonious resource allocation in context-aware n-version programming. In: Proceedings of the 7th IET System Safety Conference. The Institute of Engineering and Technology (2012) 9, 11
9. Colten, C.E., Kates, R.W., Laska, S.B.: Community resilience: Lessons from new orleans and hurricane katrina. Tech. Rep. 3, Community and Regional Resilience Institute (CARRI) (2008) 5
10. De Florio, V.: On the role of perception and apperception in ubiquitous and pervasive environments. In: Proceedings of the 3rd Workshop on Service Discovery and Composition in Ubiquitous and Pervasive Environments (SUPE'12) (2012). DOI http://dx.doi.org/10.1016/j.procs.2012.06.172. URL http://www.sciencedirect.com/science/article/pii/S1877050912005297 2, 3
11. De Florio, V.: On the constituent attributes of software and organisational resilience. Interdisciplinary Science Reviews **38**(2) (2013). URL http://www.ingentaconnect.com/content/maney/isr/2013/00000038/00000002/art00005 4
12. De Florio, V.: Preliminary contributions towards auto-resilience. In: Proceedings of the 5th International Workshop on Software Engineering for Resilient Systems (SERENE 2013), Lecture Notes in Computer Science 8166, pp. 141–155. Springer, Kiev, Ukraine (2013) 3, 4
13. De Florio, V.: Antifragility = elasticity + resilience + machine learning. Models and algorithms for open system fidelity. Procedia Computer Science **32**, 834–841 (2014). DOI http://dx.doi.org/10.1016/j.procs.2014.05.499. URL http://www.sciencedirect.com/science/article/pii/S1877050914006991. 1st ANTIFRAGILE workshop (ANTIFRAGILE-2015), the 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014) 6
14. De Florio, V.: Behavior, organization, substance: Three gestalts of general systems theory. In: Proceedings of the IEEE 2014 Conference on Norbert Wiener in the 21st Century. IEEE (2014) 2, 5
15. De Florio, V.: On the Behavioral Interpretation of System-Environment Fit and Auto-Resilience. In: Proc. of the IEEE 2014 Conference on Norbert Wiener in the 21st Century. IEEE (2014) 2, 3, 4
16. De Florio, V.: Fractally-organized connectionist networks: Conjectures and preliminary results. In: F. Daniel, O. Diaz (eds.) Proc. of the 1st Workshop on PErvasive WEb Technologies, trends and challenges (PEWET 2015), ICWE 2015 Workshops, 15th International Conference on Web Engineering (ICWE 2015), *Lecture Notes in Computer Science*, vol. 9396, pp. 1–12. Springer Int.l Publishing, Switzerland (2015). DOI 10.1007/978-3-319-24800-4_5 2, 5
17. De Florio, V.: On resilient behaviors in computational systems and environments. Journal of Reliable Intelligent Environments pp. 1–14 (2015). DOI 10.1007/s40860-015-0002-6. URL http://dx.doi.org/10.1007/s40860-015-0002-6 2, 6, 11
18. De Florio, V.: On the quality of emergence in complex collective systems. CoRR **abs/1506.01821** (2015). URL http://arxiv.org/abs/1506.01821 2
19. De Florio, V.: Reflections on organization, emergence, and control in sociotechnical systems. In: R. MacDougall

Proceedings of the 5th ACM International Conference on Distributed Event-Based Systems (DEBS), pp. 63–74. Association for Computing Machinery, Inc. (ACM) (2011). DOI http://dx.doi.org/10.1145/2002259.2002271 9, 11

(ed.) Communication and Control: Tools, Systems, and New Dimensions. Lexington (2015). URL https://arxiv.org/abs/1412.6965 2

20. De Florio, V., Bakhouya, M., Coronato, A., Di Marzo Serugendo, G.: Models and concepts for socio-technical complex systems: Towards fractal social organizations. Systems Research and Behavioral Science **30**(6) (2013) 2, 8, 9

21. De Florio, V., Blondia, C.: Service-oriented communities: Visions and contributions towards social organizations. In: R. Meersman, T. Dillon, P. Herrero (eds.) On the Move to Meaningful Internet Systems: OTM 2010 Workshops, *Lecture Notes in Computer Science*, vol. 6428, pp. 319–328. Springer Berlin / Heidelberg (2010). DOI http://dx.doi.org/10.1007/978-3-642-16961-8_51 2, 8

22. De Florio, V., Coronato, A., Bakhouya, M., Di Marzo Serugendo, G.: Service-oriented communities: Models and concepts towards fractal social organizations. In: Proceedings of the 8th International Conference on signal, image technology and internet based systems (SITIS 2012). IEEE (2012) 2

23. De Florio, V., Pajaziti, A.: How resilient are our societies? analyses, models, and preliminary results. In: Complex Systems (WCCS), 2015 Third World Conference on. IEEE (2015) 2, 12

24. De Florio, V., Pajaziti, A.: Tapping into the wells of social energy: A case study based on falls identification. CoRR **abs/1508.06655** (2015). URL http://arxiv.org/abs/1508.06655 3, 12

25. De Florio, V., Primiero, G.: A framework for trustworthiness assessment based on fidelity in cyber and physical domains. Procedia Computer Science **52**, 996–1003 (2015). DOI http://dx.doi.org/10.1016/j.procs.2015.05.092. URL http://www.sciencedirect.com/science/article/pii/S1877050915008923. Proc. of the 2nd Workshop on Computational Antifragility and Antifragile Engineering (ANTIFRAGILE'15), in the framework of the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015) 6

26. De Florio, V., Sun, H., Blondia, C.: Community resilience engineering: Reflections and preliminary contributions. In: I. Majzik, M. Vieira (eds.) Software Engineering for Resilient Systems, *Lecture Notes in Computer Science*, vol. 8785, pp. 1–8. Springer International Publishing (2014). DOI 10.1007/978-3-319-12241-0_1. URL http://dx.doi.org/10.1007/978-3-319-12241-0_1 2

27. De Florio, V., Sun, H., Buys, J., Blondia, C.: On the impact of fractal organization on the performance of socio-technical systems. In: Proceedings of the 2013 International Workshop on Intelligent Techniques for Ubiquitous Systems (ITUS 2013). IEEE, Vietri sul Mare, Italy (2013) 2

28. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning about a Highly Connected World, chap. Games, pp. 155–208. Cambridge University Press (2010) 6

29. Froehlich, T.: Persistence of vision raytracer on-line documentation repository. http://wiki.povray.org/content/Documentation:Contents (2012). [Online; accessed 19-July-2014] 10

30. Huynh, Q.T., Nguyen, U.D., Tran, S.V., Nabili, A., Tran, B.Q.: Fall detection system using combination accelerometer and gyroscope. In: Proc. of the Second Int.l Conf. on Advances in Electronic Devices and Circuits (EDC 2013) (2013). DOI 10.3850/978-981-07-6261-2_56 3

31. Jackson, G.B.: Contemporary Viewpoints on Human Intellect and Learning. Xlibris Corporation (2010) 2

32. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**, 41–50 (2003). DOI http://dx.doi.org/10.1109/MC.2003.1160055. URL http://dx.doi.org/10.1109/MC.2003.1160055 2

33. Koestler, A.: The Ghost in the Machine. Macmillan (1967) 7, 8

34. Latour, B.: On actor-network theory. a few clarifications plus more than a few complications. Soziale Welt **47**, 369–381 (1996) 4

35. Law, J.: Notes on the theory of the actor-network: Ordering, strategy and heterogeneity. Systems Practice (5), 379–393 (1992) 8

36. Leibniz, G., Strickland, L.: The shorter Leibniz texts: a collection of new translations. Continuum impacts. Continuum (2006). URL http://books.google.be/books?id=oFoCY3xJ8nkC 3, 6

37. Lycan, W.: Consciousness and experience. Bradford Books. MIT Press (1996) 4

38. Meystre, S.: The current state of telemonitoring: a comment on the literature. Telemed J E Health **11**(1), 63–69 (2005). URL http://www.ncbi.nlm.nih.gov/pubmed/15785222 7

39. Miskel, J.F.: Disaster Response and Homeland Security: What Works, What Doesn't. Stanford University Press, Stanford, CA (2008). URL http://www.sup.org/books/title/?id=16470 11

40. Nienhuys, H.W., Nieuwenhuizen, J., Reuter, J., Zedeler, R.: GNU LilyPond — The Music Typesetter. GNU Free Documentation (2004). For LilyPond version 2.2.5. Available from http://www.lilypond.org 6

41. Nilsson, T.: How neural branching solved an information bottleneck opening the way to smart life. In: Proceedings of the 10th International Conference on Cognitive and Neural Systems. Boston University, MA (2008) 3

42. Nilsson, T.: Solving the sensory information bottleneck to central processing in complex systems. In: A. Yang, Y. Shan (eds.) Intelligent Complex Adaptive Systems, pp. 159–186. IGI Global, Hershey, PA (2008). DOI 10.4018/978-1-59904-717-1.ch006 3

43. Nilsson, T.: Spatial multiplexing: Solving information bottlenecks in real neural systems and the origin of brain rhythms. International Journal of Adaptive, Resilient and Autonomic Systems pp. 46–70 (2014). DOI 10.4018/ijaras.2014100104 3

44. OASIS: Web services base notification 1.3 standard. Tech. rep., OASIS (2006). URL http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf 7

45. OASIS: Web services service group 1.2 standard. Tech. rep., OASIS (2006). URL http://docs.oasis-open.org/wsrf/wsrf-ws_service_group-1.2-spec-os.pdf 7

46. RAND: Community resilience (2014). Available at URL http://www.rand.org/topics/community-resilience.html 5

47. Rosenblueth, A., Wiener, N., Bigelow, J.: Behavior, purpose and teleology. Philosophy of Science **10**(1), 18–24 (1943). URL http://www.journals.uchicago.edu/doi/abs/10.1086/286788 1, 2, 3, 5, 6

48. Sharp, J.A. (ed.): Data Flow Computing: Theory and Practice. Ablex Publishing Corp., Norwood, NJ, USA (1992) 8

49. Sprugnoli, R.: Algebra moderna e matematiche finite. ETS, Pisa, Italy (1979) 2

50. Stokols, D., Lejano, R.P., Hipp, J.: Enhancing the resilience of human-environment systems: a social ecological perspective. Ecology and Society **18**(1), 7 (2013) 11

51. Sun, H., De Florio, V., Blondia, C.: Implementing a role based mutual assistance community with semantic service description and matching. In: Proc. of the Int.l Conference on Management of Emergent Digital EcoSystems (MEDES) (2013) 8

52. Sun, H., De Florio, V., Gui, N., Blondia, C.: Participant: A new concept for optimally assisting the elder people. In: Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems (CBMS-2007). IEEE Computer Society, Maribor, Slovenia (2007) 11

53. Taleb, N.N.: Antifragile: Things That Gain from Disorder. Random House Publishing Group (2012) 11

54. Tëmkin, I., Eldredge, N.: Networks and hierarchies: Approaching complexity in evolutionary theory. In: E. Serrelli, N. Gontier (eds.) Macroevolution: Explanation, Interpretation, Evidence, pp. 183–226. Springer International Publishing (2015). DOI 10.1007/978-3-319-15045-1 1, 5

55. Tharumarajah, A., Wells, A.J., Nemes, L.: Comparison of the bionic, fractal and holonic manufacturing system concepts. International Journal Computer Integrated Manufacturing (1996) 7, 8

56. Tharumarajah, A., Wells, A.J., Nemes, L.: Comparison of emerging manufacturing concepts. In: Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, vol. 1, pp. 325–331 vol.1 (1998). DOI 10.1109/ICSMC.1998.725430 7, 8

57. Tomasulo, R.M.: IBM Journal of Research and Development pp. 25–33 (1967) 8

58. Wagner, G.P., Altenberg, L.: Perspective: Complex adaptations and the evolution of evolvability. Evolution **50**(3) (1996). URL http://dynamics.org/Altenberg/FILES/GunterLeeCAEE.pdf 5

59. Warnecke, H., Hüser, M.: The fractal company: a revolution in corporate culture. Springer (1993). URL http://books.google.be/books?id=YsQSAQAAMAAJ 7, 8

60. Washburn, J.: Post to the "Philosophy of Zappa" facebook group (2015). URL https://goo.gl/JhMjwI 6

61. Weddell, G., Taylor, D., Williams, C.: The patterned arrangement of spinal nerves to the rabbit ear. J. Anat. **89**(part 3), 317–342 (1955) 3

62. Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: A review. Pervasive and Mobile Computing **8**(1), 36–66 (2012) 7, 9